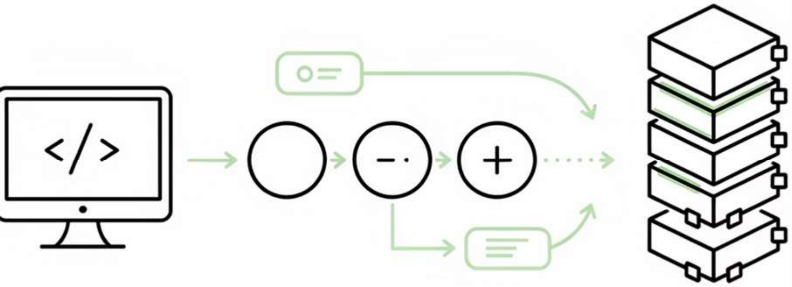


Docker Login Using GitHub Secrets

Secure authentication and deployment strategies for containerized applications

Docker Login Using GitHub Secrets

```
- name: Login to Docker Hub
  uses: docker/login-action@v3
  with:
    username: ${{ secrets.DOCKER_USERNAME }}
    password: ${{ secrets.DOCKER_PASSWORD }}
```

Security Explanation

- Secrets are encrypted
- Not visible in logs
- Scoped to repository

GitHub Secrets: Best Practices

✅ *Use secrets for:*

- Registry credentials
- Tokens
- API keys

❌ *Never:*

- Hardcode passwords
- Echo secrets in logs
- Commit .env files

Docker Build & Push Step

```yaml
- name: Build and Push Image
  uses: docker/build-push-action@v5
  with:
    context: .
    push: true
    tags: vishwacloudlab/python-app:latest
```

Explanation

context → build directory

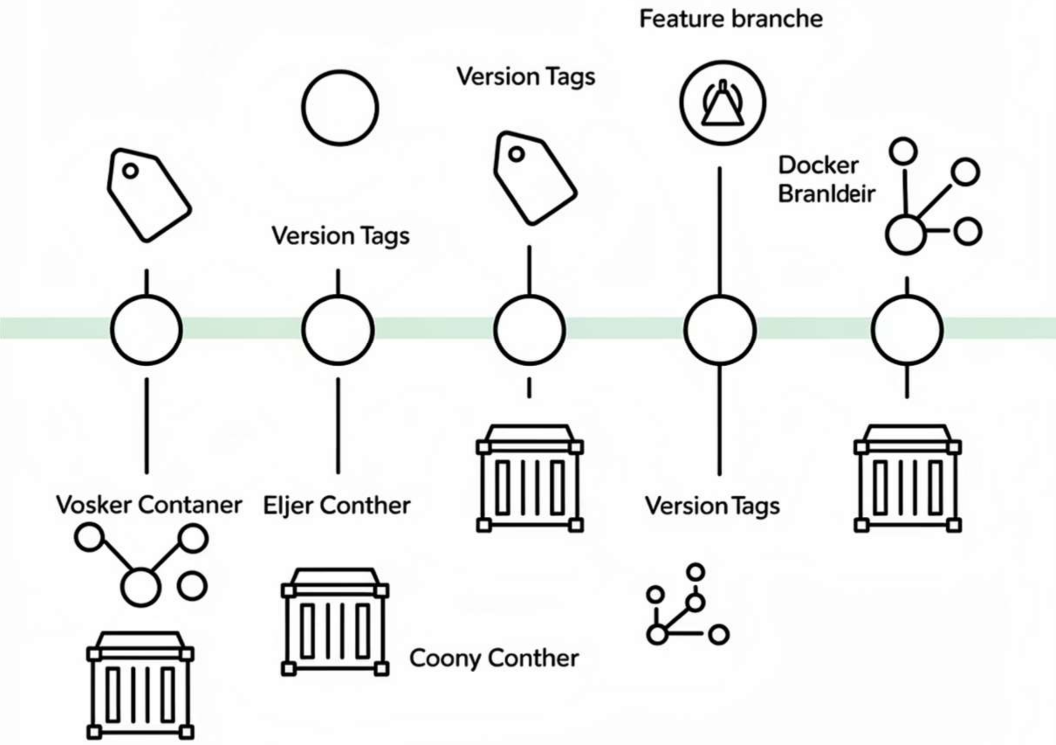
push: true → sends image to registry

tags → image naming

Versioning Images Using Git Tags

Why Version Images?

- Rollback support
- Traceability
- Release consistency



Git Tag-Based Workflow Trigger

```
on:
  push:
    tags:
      - 'v*.*.*'
```

Example Tags

- v1.0.0
- v1.2.3

Using Git Tag as Image Version

```
tags: |
  vishwacloudlab/python-app:${{ github.ref_name }}
```

Result

`vishwacloudlab/python-app:v1.0.0`

> 📌 Enterprise Best Practice:
>
> Image version = Git tag = Release version

Full CI Workflow (Consolidated)

```
name: Docker Release
on:
 push:
 tags:
 - 'v*.*.*'
jobs:
 release:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v4
 - uses: docker/setup-buildx-action@v3
 - uses: docker/login-action@v3
 with:
 username: ${{ secrets.DOCKER_USERNAME }}
 password: ${{ secrets.DOCKER_PASSWORD }}
 - uses: docker/build-push-action@v5
 with:
 push: true
 tags: vishwacloudlab/python-app:${{ github.ref_name }}
```

Basic Release Strategies

1

Latest Only

- Always overwrite latest
- ❌ No rollback

2

Versioned Releases

- v1.0.0, v1.0.1
- ✅ Safe rollback

3

Environment-Based Tags

- dev, qa, prod
- Common in enterprises

Release Strategy Mapping

Strategy	Tag Example	Use Case
Latest	latest	Dev testing
Semantic	v1.2.0	Production
Env	prod	Environment deploy

Common Mistakes & Fixes

Mistakes

Mistake	Fix
Hardcoded credentials	Use GitHub Secrets
No versioning	Use Git tags
Large images	Slim base images
No rollback	Versioned tags

CI/CD Security Best Practices

- Least-privilege secrets
- Rotate credentials
- Scan images (Trivy, Snyk)
- Use protected branches
- Enforce PR checks